Unit 4: Problems

Problem 1

An M-matrix of size FxC is available (F is the number of rows and C the number of columns), whose cells have a positive integer numeric value. For example, the matrix with 4 rows and 5 columns below:

3	2	3	2	5
1	2	4	5	6
4	3	6	2	4
2	5	3	4	3

A movement between the squares M_{ij} and M_{pq} is valid only if (p=i && q=j+1) or if (p=i+1 && q=j). The path of the matrix is defined as the sequence of movements that lead from box M_{11} to the box M_{FC} and the cost of a path is equal to the sum of the values of the boxes that it crosses.
Design a Dynamic Programming algorithm that obtains the lowest cost of all the paths of a given matrix.

Problem 2

You want to get the first N prime numbers (for example, if N = 5 you would have to get the primes 2, 3, 5, 7 and 11). Design a Dynamic Programming algorithm that solves the problem, finding a way to reuse the results already achieved to calculate the new prime numbers.

Problem 3

There is a system of banknotes of different values and ordered from lowest to highest (for example 1, 2, 5, 10, 20, 50 and 100 euros), which are represented by the values v_i, with $i \in \{1, \dots, N\}$ (in the previous case, N = 7) so that each banknote has a finite amount, greater or equal to zero, which is saved in c_i (following the example, c_3=6 would represent that there are 6 banknotes of 5 euros).
You want to pay exactly a certain amount of money D, using the least amount of banknotes possible. It is known that $D \leq \sum_{i=1}^{n} c_i v_i$, but the exact amount D may not be obtainable through the available banknotes.
Design an algorithm with the Dynamic Programming methodology that determines, having as data the values c_i, v_i and D:

- if the amount D can be returned exactly or not, and
- in the affirmative case, how many notes of each type form the optimum decomposition.

Problem 4

Ali Baba has managed to enter the cave of the one hundred and one thousand thieves, and has brought his camel along with two large panniers. The problem is that he finds so much treasure that he does not know what to take. The treasures are carved jewels, works of art, ceramics ... that is, they are unique objects that cannot be split since then their value would be reduced to zero.

Fortunately the thieves have everything very well organized and find a list of all the treasures in the cave, which reflects the weight of each piece and its value in the Damascus market. For his part, Ali knows the weight capacity of each of the saddlebags.
Design an algorithm that, taking as data the weights and value of the pieces, and the capacity of the two saddlebags, allows to obtain the maximum benefit that Ali Baba can get from the cave of wonders.

Problem 5
We have a directed graph G=<N, A>, where N={1,..., n} is the set of nodes and E⊆NxN is the set of edges. Let M be the adjacency matrix of the graph G, that is, M[i, j]=TRUE if the edge (i, j) exists, and M[i, j]=FALSE otherwise.
You are interested in knowing from which vertices you can access any other vertex (using a path of any length), using the Warshall algorithm, so that

- It is necessary to obtain a matrix of paths C so that C[i, j]=TRUE if there is a path (of any length) between the nodes i and j, and C[i, j]=FALSE if there is no way to get from i to j.
- To do this, it is necessary to consider a higher number of nodes with a Bottom-Up approach. You start by trying to go directly from one node to another, then you try to find the paths that can use the vertex 1, then those that can use the vertices 1 and 2, then what you get with the vertices 1, 2 and 3, etc., until you get the paths that can use all vertices from 1 to n.

Implement a Dynamic Programming method in the above terms, which obtains the matrix C of existence of paths of a graph having as data the number of nodes n and the adjacency matrix M.

Problem 6
The EscobaBall tournament is here, and wilder than ever! This year, in the College of Magic and Sorcery have decided that the four teams (Griffins, Snakes, Ravens and Badgers) play in each game all against all, and as long as no match ends in a draw. The tournament will end when the same team has won a total of N matches (not necessarily consecutive).
Wizard apprentice Javi Potter wants to bet some money for his team, the Griffins, so he goes to the gnomes' betting house to see how much they would give him if his team wins: his winnings would be equal to the amount of money wagered divided by the probability that the team wins the championship (for example, if the Griffins had a 50% winning the championship and Javi bets 10 gold coins, his possible winnings would be 10/0.5=20 gold coins; had a 20% to win, the possible benefit would be 10/0.2=50 gold coins, greater reward because it is more difficult to achieve).
To obtain this probability, the bookmaker has a Quality Value assigned to each team (which measures the skill of the players, their motivation, etc., and that is a fixed value for the team and independent of the match that is playing). so that the higher the VC of a team, the more likely it is to win a game. For example, if all four teams had the same VC they would all have a 25% chance of winning a game. If three teams had the same VC and the fourth team had twice that amount, the first would have 20% and the last 40%. As the matches cannot end in a draw, the sum of the odds is always 100%.
Taking as data the Quality Values of the teams, the number of N matches that a team must win to win the tournament, and the money D wagered by Javi Potter, obtain what the winnings would be if the Griffins win.

Problem 7

A sequence of bits A is defined as a sequence A={a_1, a_2,..., a_n} where each a_i can take the value 0 or the value 1, and n is the length of the sequence A. From a sequence it is defined a subsequence X of A as X = {x_1, x_2,..., x_k}, where $k \leq n$, so that X can be obtained by eliminating some element of A but respecting the order in which the bits appear; for example, if A={1,0,1,1,0,0,1} we could obtain as subsequences {1,1,1,0,1}, {1,0,1} or {1,1,0,0} among others, but you could never get the subsequence {1,0,0,1,1}.

Given two sequences A and B, X is called a common subsequence of A and B when X is a subsequence of A and is also a subsequence of B. (although they may have been obtained by removing different elements in A than B, and even different quantities of elements). Assuming the sequences A = {0,1,1,0,1,0,1,0} and B = {1,0,1,0,0,1,0,0,1}, a common subsequence would be X = {1,1,0,1}, but it could not be X = {0,1,1,1,0}.

We want to determine the common subsequence of two sequences A and B that have the maximum length, for which it is requested

- explain in detail how to solve the problem, and
- make a Dynamic Programming algorithm that obtains the maximum possible length and a common sequence of that length.

Deliverables: a problem to choose among problems 3 and 4 and a problem to choose among problems 6 and 7.